

# **Neural Net studies for EM/Photon ID**

Alex Melnitchouk

**University of Mississippi**

Dhiman Chakraborty

**Northern Illinois University**

**09/20/2004 CALGO Group Meeting**



# Neural Net (NN) studies

1. Started with the help from Yurii Maravin, who initiated these studies with a focus on  $W\gamma$  analysis
2. Goal: reduce background relative to current EMID cuts keeping the same efficiency

Why can we possibly expect improvement from NN in comparison with the HMatrix ?

- NN can resolve non-linear correlations between input variables while HMatrix can not  
*(NN is not expected to make any difference if input variables are linearly correlated )*
- Easier to use. Due to available implementations NN training/using cycles are much faster than those for HMatrix: can try exploring different sets of input variables

# Documentation

latest  $h \rightarrow \gamma\gamma$  results that called for NN studies:

[http://www-d0.fnal.gov/~melnit/NN/h.June\\_17\\_2004.pdf](http://www-d0.fnal.gov/~melnit/NN/h.June_17_2004.pdf)

first attempts of NN studies:

[http://www-d0.fnal.gov/~melnit/NN/nn.July\\_01\\_2004.pdf](http://www-d0.fnal.gov/~melnit/NN/nn.July_01_2004.pdf)

Talk at Higgs meeting July 08 2004

[http://www-d0.fnal.gov/~melnit/NN/nn.July\\_15\\_2004.pdf](http://www-d0.fnal.gov/~melnit/NN/nn.July_15_2004.pdf)

[http://www-d0.fnal.gov/~melnit/NN/nn.July\\_22\\_2004.pdf](http://www-d0.fnal.gov/~melnit/NN/nn.July_22_2004.pdf)

(also available at Higgs-Dilepton group meeting agenda)

# NN description

- object oriented implementation of MLPfit package available in ROOT via **TMultilayerPerceptron** class
- ROOT version = 4.00/04, D0 Run II release = t04.04.00
- simple feed-forward network:
  - input nodes
    - renormalize input:  
 $\text{input } i \rightarrow \text{input } i = [\text{input}^i - \text{Mean}] / \text{RMS}$
    - hidden nodes (one layer)
      - use sigmoid transfer functions  $g(x) = 1/[1+\exp(-x)]$
      - where  $x = \text{node threshold} + \text{sum of weighted inputs}$
    - output node is linear i.e.  $g(x) = x$
  - the code for the trained NN can be written out (as a class) in the output source and header files to be used “offline” in the analysis

# Learning Method

- Use Broyden, Fletcher, Goldfarb, Shanno (BFGS) learning method.
- Parameters are **RESET** and **TAU**.
- **RESET** = number of epochs after which the line search direction is reset to the steepest descent (default=50)
- **TAU**: lower value = higher precision, slower search (default = 3.0)

# **Number of training epochs and hidden nodes**

- Number of training epochs = 800 (on the plateau,  
see earlier notes)
- Number of hidden nodes = twice the number  
of input nodes

# Input Variables

- As a first step use the same variables that are used by HMatrix7  
*(keeping in mind that it is not optimal in terms of electron vs. photon efficiency – Yurii’s studies)*
- Use HMatrix7 variables + track isolation as input variables to NN and compare NN performance with HMatrix7 + Track Isolation EMID
- HMatrix7 variables are:  
EM floor energy fractions (4 variables) ,  $r\phi$ -width in EM3  
 $\log(\text{energy})$ ,  $Z(\text{Primary Vertex})/\sigma$ .
- Do not use  $Z(\text{Primary Vertex})/\sigma$  for NN  
(no  $\eta$ -binning for NN)  $\Rightarrow$  7 variable NN

# Choice of Training Samples

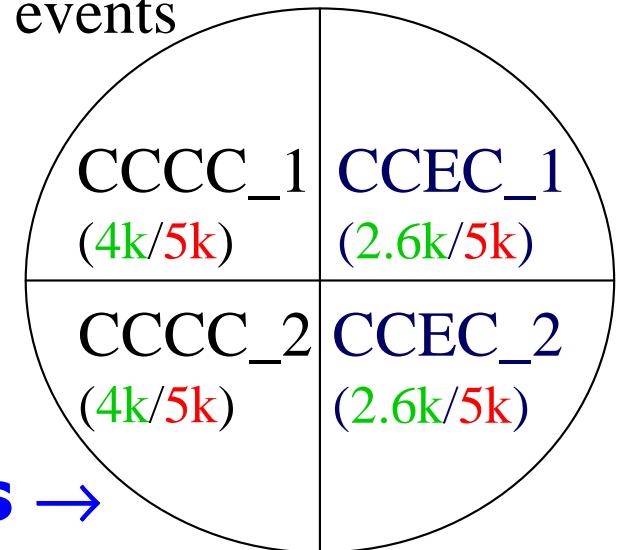
- use real data for NN training
  - ignore for the moment the difference in efficiency between photons and electrons
    - .i.e. assume for the moment that calorimeter-wise Electron = Photon = EM object
  - consider only CC objects
  - distinguish between CC objects coming from CCCC events and those from CCEC events
- (this is purely  $h \rightarrow \gamma\gamma$  driven, see e.g. [http://www-d0.fnal.gov/~melnit/NN/h.June\\_17\\_2004.pdf](http://www-d0.fnal.gov/~melnit/NN/h.June_17_2004.pdf) )*

# NN Training Samples

1. NN Signal Sample :  $Z \rightarrow ee$  data (double track match)  
from CS 2EMhighpt skim, v6 d0correct.  
This is electron data sample of >99% electron purity
  
2. NN Background Sample : Single(exactly one)EM data  
from CS 1EMloose skim, v6 d0correct.  
This is mixed data sample ( $j$  and  $\gamma$  from  $jj$ ,  $j\gamma$ ,  $\gamma\gamma$ )  
of >86% jet purity in  $M(jj)=[100-150]$  GeV

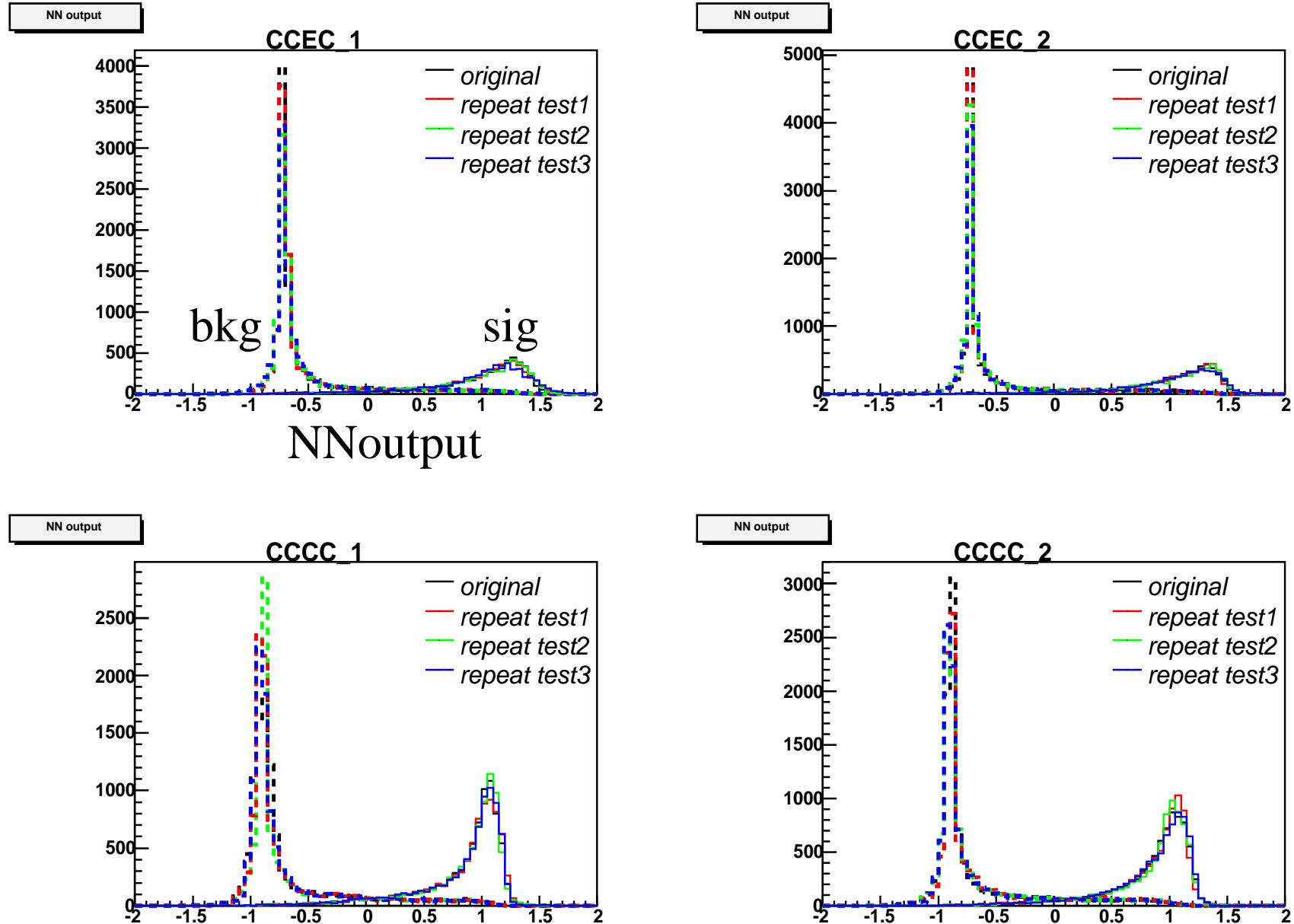
Select CC objects from CCCC and CCEC events

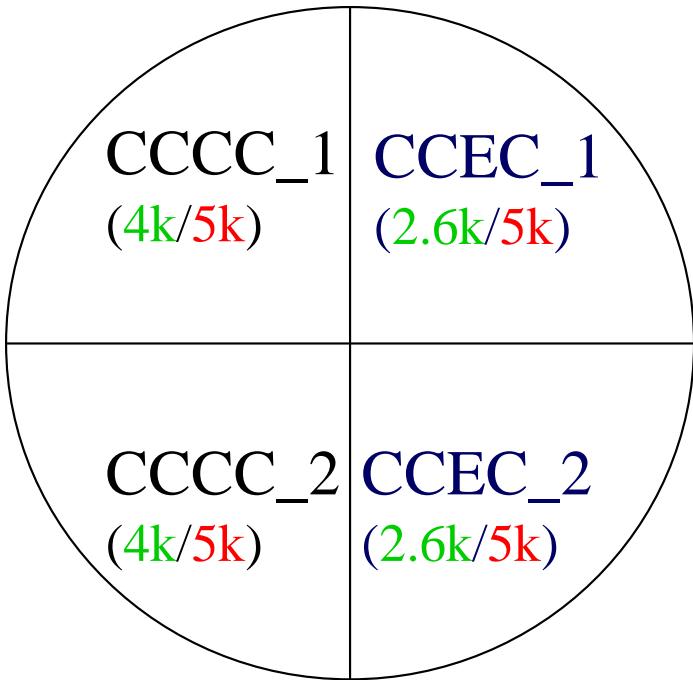
- *CC objects near  $\phi$ -module boundaries are accepted,*
- *require isolation <0.15,*
- *kinematic cuts:  $PT>25$  GeV*  
*Mass=[40,160] GeV*



**use 4 orthogonal sig/bkg samples →**

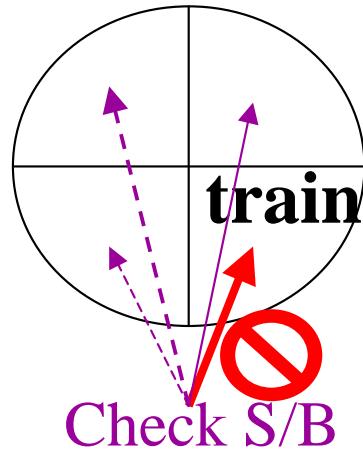
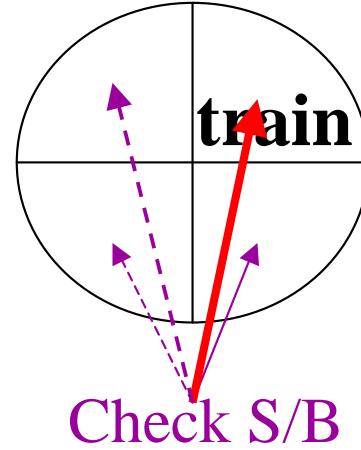
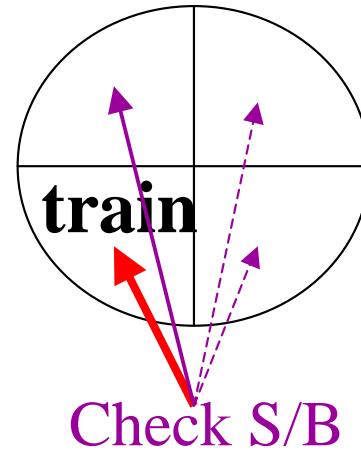
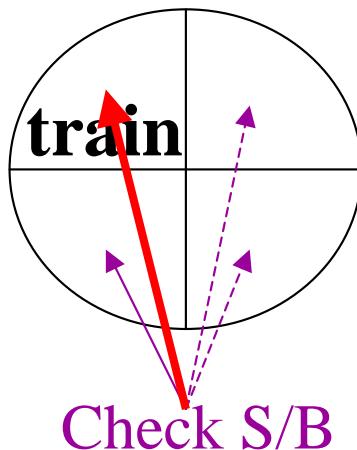
# How much do the weights fluctuate? repeat full training several times, no parameter changes

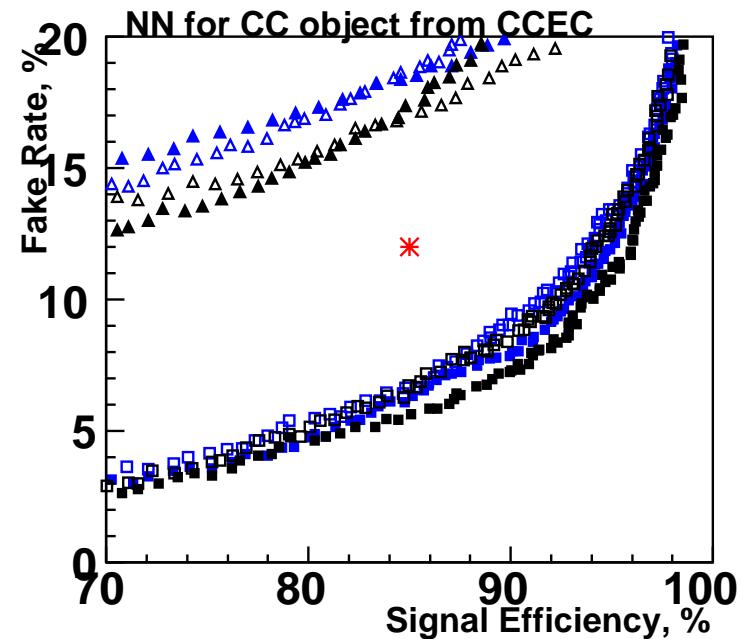
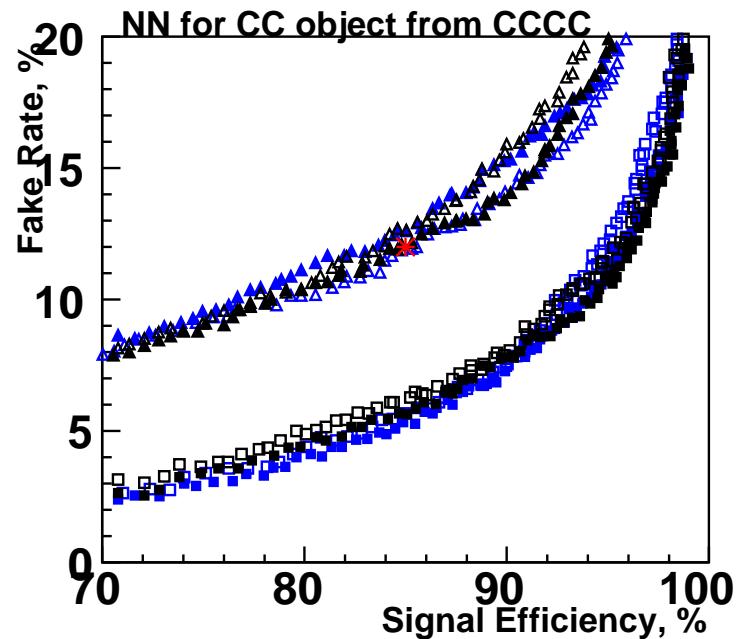




**4 orthogonal sig/bkg samples:**

**Calculate**  
**Signal Efficiency vs. Fake Rate**  
**for all**  
**TrainSample – OutputSample pairs**





## NN S/B performance

- \* current EMID
- out: CCCC\_1, train: CCCC\_1
- out: CCCC\_1, train: CCCC\_2
- ▲ out: CCCC\_1, train: CCEC\_1
- △ out: CCCC\_1, train: CCEC\_2
- out: CCCC\_2, train: CCCC\_2
- out: CCCC\_2, train: CCCC\_1
- ▲ out: CCCC\_2, train: CCEC\_2
- △ out: CCCC\_2, train: CCEC\_1

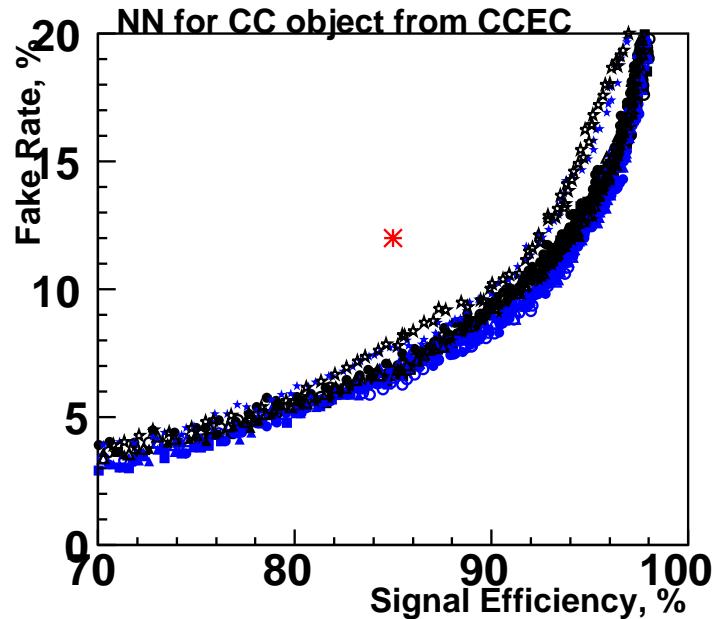
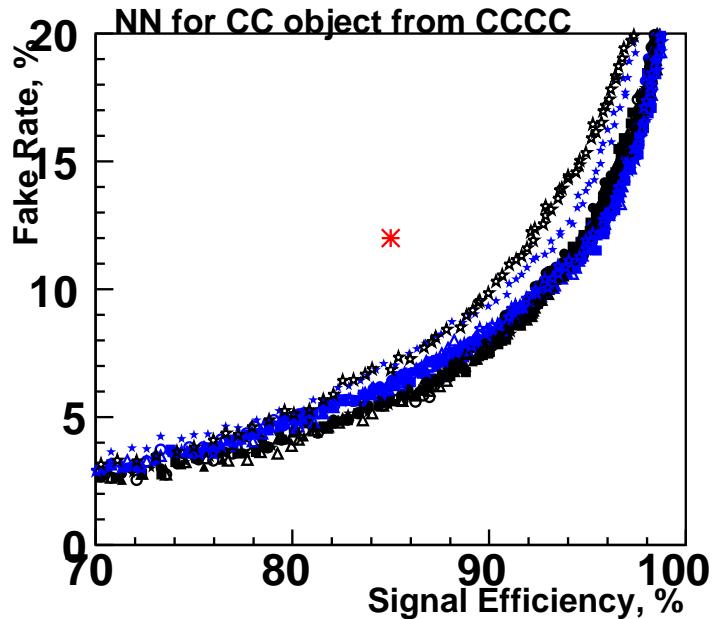
- \* current EMID
- out: CCEC\_1, train: CCEC\_1
- out: CCEC\_1, train: CCEC\_2
- ▲ out: CCEC\_1, train: CCCC\_1
- △ out: CCEC\_1, train: CCCC\_2
- out: CCEC\_2, train: CCEC\_2
- out: CCEC\_2, train: CCEC\_1
- ▲ out: CCEC\_2, train: CCCC\_2
- △ out: CCEC\_2, train: CCCC\_1

# Testing NN robustness

- Keep only the following TrainSample – OutputSample pairs:

CCCC\_1 – CCCC\_2 , CCCC\_2 – CCCC\_1  
CCEC\_1 – CCEC\_2 , CCEC\_2 – CCEC\_1

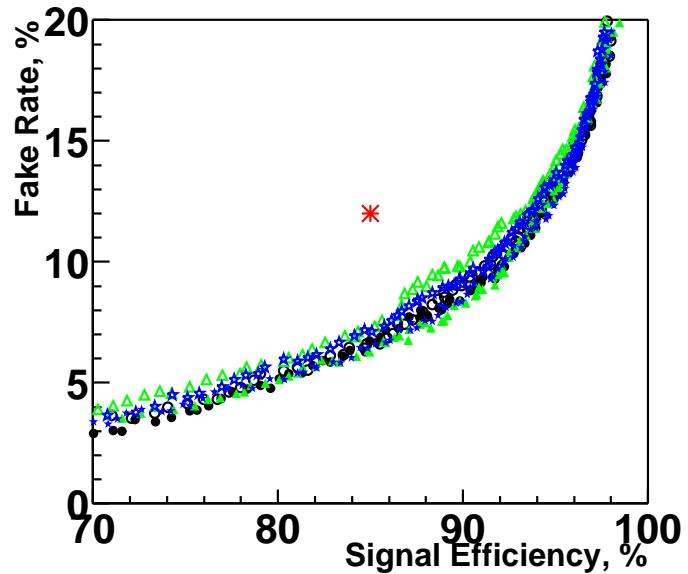
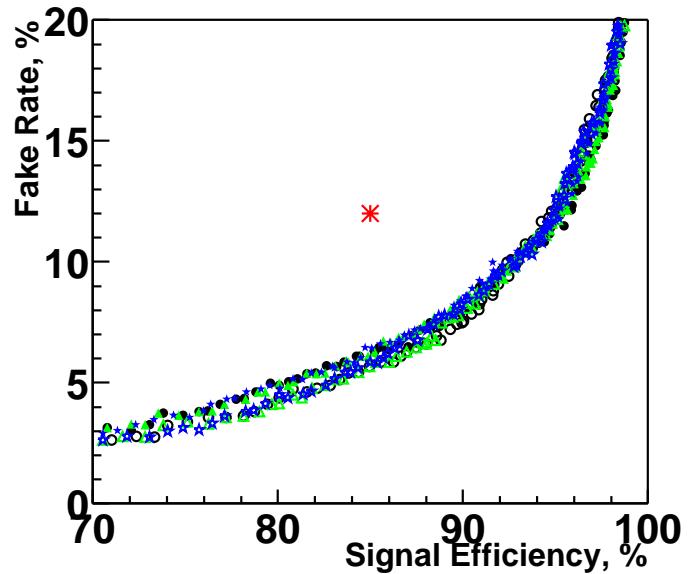
- i.e. exclude cross-topology pairs and cases where the NN has seen the sample
- Check how sensitive the NN performance is to :
  1. variation of RESET and TAU parameters
  2. reordering inputs
  3. renormalizing inputs
  4. variation of the number of hidden nodes



## NN robustness test: parameters, reordering inputs, renormalizing inputs

- \* current EMID
- par=50,3.0(DFLT) | out: 2, train: 1
- par=50,3.0(DFLT) | out: 1, train: 2
- par=50,4.0 | out: 2, train: 1
- par=50,2.0 | out: 2, train: 1
- par=50,4.0 | out: 1, train: 2
- par=50,2.0 | out: 1, train: 2
- ▲ par=25,4.0 | out: 2, train: 1
- △ par=25,2.0 | out: 2, train: 1
- ▲ par=25,4.0 | out: 1, train: 2
- △ par=25,2.0 | out: 1, train: 2
- \* reorder inputs | out: 1, train: 2
- ★ reorder inputs | out: 2, train: 1
- ★ sigmoid inputs | out: 2, train: 1
- ★ sigmoid inputs | out: 1, train: 2

- \* current EMID
- par=50,3.0(DFLT) | out: 2, train: 1
- par=50,3.0(DFLT) | out: 1, train: 2
- par=50,4.0 | out: 2, train: 1
- par=50,2.0 | out: 2, train: 1
- par=50,4.0 | out: 1, train: 2
- par=50,2.0 | out: 1, train: 2
- ▲ par=25,4.0 | out: 2, train: 1
- △ par=25,2.0 | out: 2, train: 1
- ▲ par=25,4.0 | out: 1, train: 2
- △ par=25,2.0 | out: 1, train: 2
- \* reorder inputs | out: 1, train: 2
- ★ reorder inputs | out: 2, train: 1
- ★ sigmoid inputs | out: 2, train: 1
- ★ sigmoid inputs | out: 1, train: 2



## NN robustness test: $2N_{\text{input}}$ , $N_{\text{input}}+2$ , $2N_{\text{input}}-1$ hidden nodes

\* current EMID

- $N_h = 2N_i$  | out: 2, train: 1
- $N_h = 2N_i$  | out: 1, train: 2
- ▲  $N_h = N_i+2$  | out: 2, train: 1
- ▲  $N_h = N_i+2$  | out: 1, train: 2
- \*  $N_h = 2N_i-1$  | out: 2, train: 1
- \*  $N_h = 2N_i-1$  | out: 1, train: 2

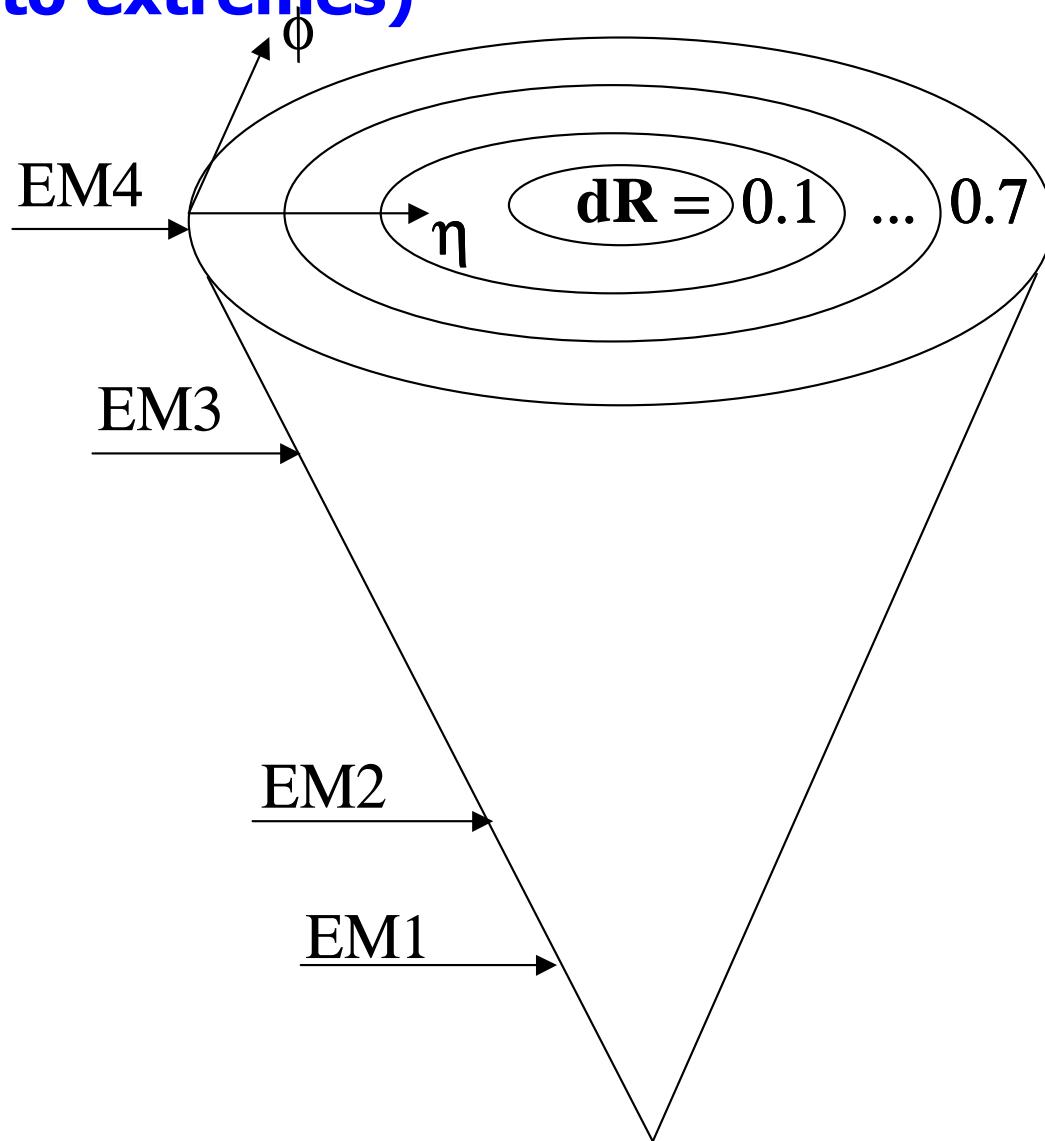
\* current EMID

- $N_h = 2N_i$  | out: 2, train: 1
- $N_h = 2N_i$  | out: 1, train: 2
- ▲  $N_h = N_i+2$  | out: 2, train: 1
- ▲  $N_h = N_i+2$  | out: 1, train: 2
- \*  $N_h = 2N_i-1$  | out: 2, train: 1
- \*  $N_h = 2N_i-1$  | out: 1, train: 2

# Summary of current NN studies

- With the 7 variable configuration NN seems to provide an additional background rejection factor of ~1.5-2 in comparison with current EMID at the same signal efficiency
- NNs seem to pick up the difference in CC objects between CCCC and CCEC, that was also seen with standard EMID (*see e.g.* [http://www-d0.fnal.gov/~melnit/NN/h.June\\_17\\_2004.pdf](http://www-d0.fnal.gov/~melnit/NN/h.June_17_2004.pdf)). Is this a problem ? short answer = yes, long answer = maybe not, in any case this may belong outside the NN studies.
- If this is not a problem then **we have two trained NNs that were shown to be robust !!!**
- Can take 2 roads now:
  1. Find optimal set of input variables to do better than 1.5~2 in background rejection
  2. Apply NN to  $h \rightarrow \gamma\gamma$  analysis to see if it works in real life

# “detailed picture” of EM cluster (taking Yurii’s original idea to make use of cell info to extremes)

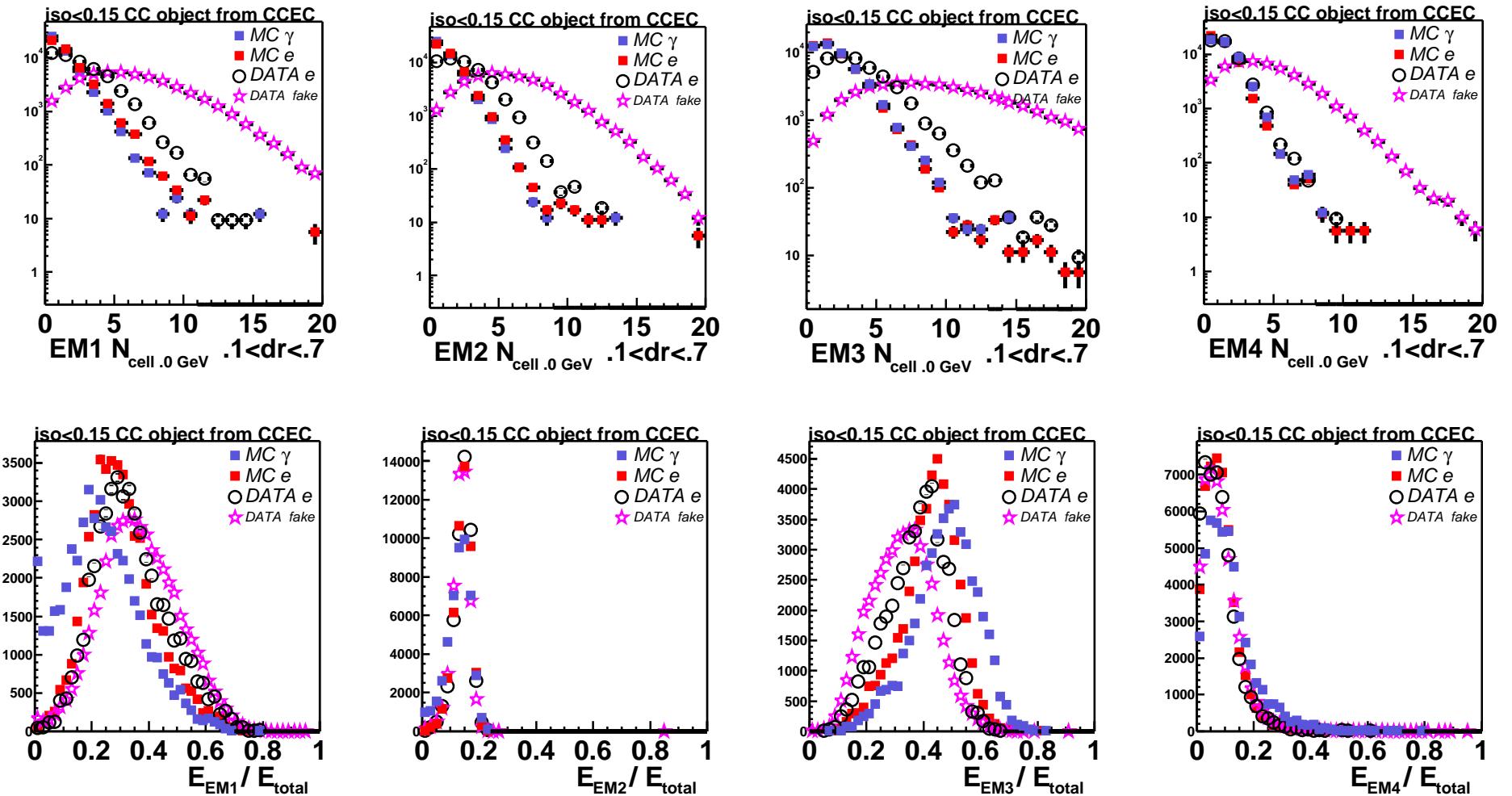


1. slice longitudinally into 4 slices corresponding to EM floors
2. subslice each of the four slices into  $N_R$  cone(shell) slices of decreasing  $dR$
3. In each of  $4 \times N_R$  regions look at :
  - number of cells
  - absolute energies
  - energy fractionsto find discriminating variables to put into NN
4. Try different  $E_{cell}$  thresholds

# List of Best Cell-level info Variables Found

1. N\_cells ( $0.1 < dR < 0.7$ ) EM1
2. N\_cells ( $0.1 < dR < 0.7$ ) EM2
3. N\_cells ( $0.1 < dR < 0.7$ ) EM3
4. N\_cells ( $0.1 < dR < 0.7$ ) EM4
5. Sum of cell energies ( $0.1 < dR < 0.7$ ) in EM1
6. Fraction of cell energy ( $0.1 < dR < 0.7$ )/  $dR < 0.7$  in EM4
7. Fraction of cell energy ( $dR < 0.1$ )/  $dR < 0.7$  in EM4

# Ncells ( $0.1 < dR < 0.7$ ) :top and EM floor energy fractions: bottom

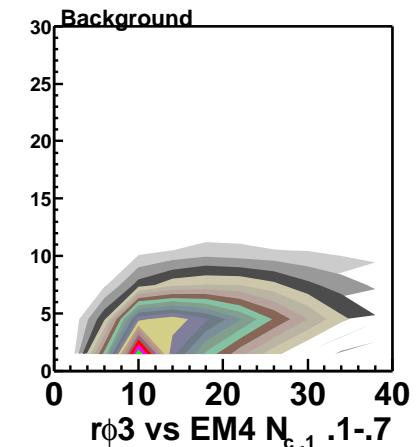
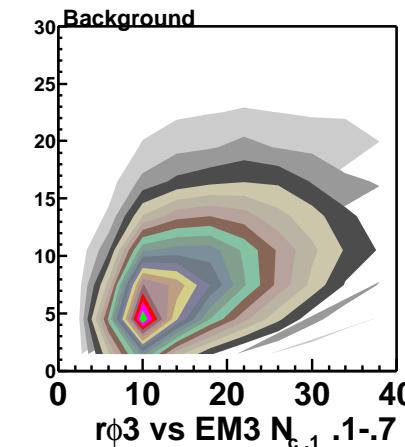
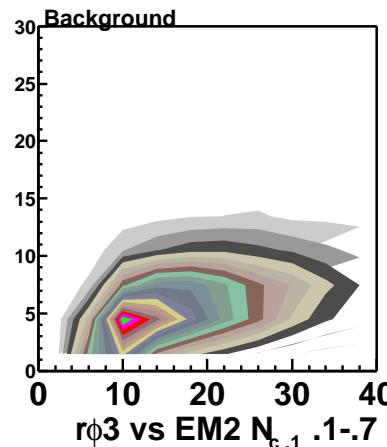
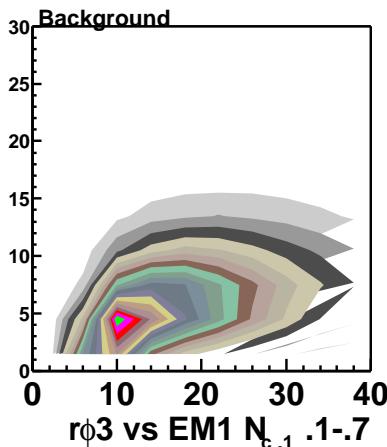
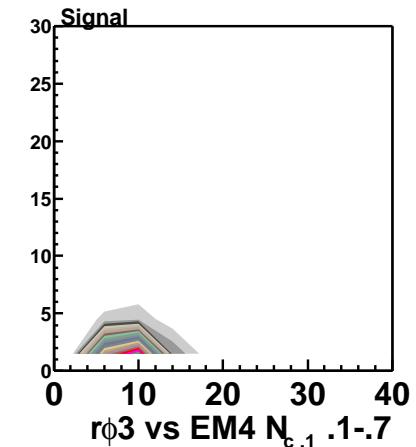
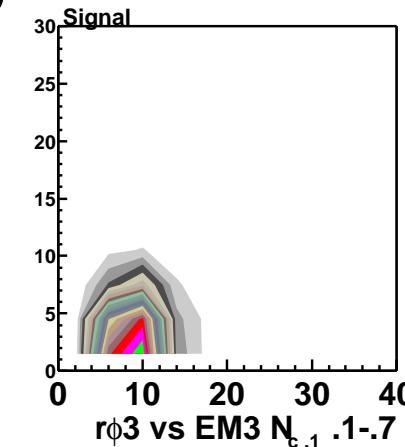
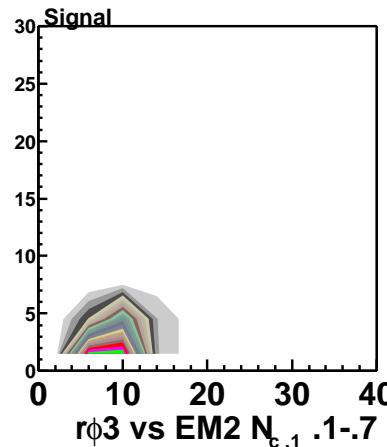
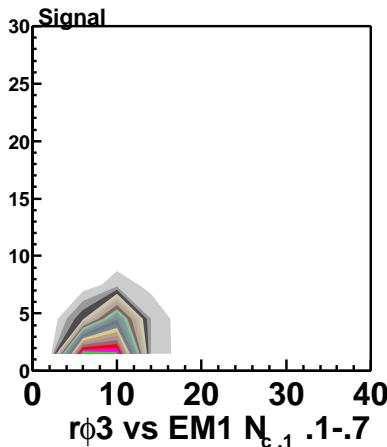


# **Full List of Variables that can potentially be used for Constructing optimal NN input configuration**

1. Rphi-width
2. Log(E)
3. Z(pvtx) (if do eta-binning)
4. EM1 energy fraction
5. EM2 energy fraction
6. EM3 energy fraction
7. EM4 energy fraction
8. Iso
9. Absiso
10. Trkiso
11. Emfrac
12. N\_cells ( $0.1 < dR < 0.7$ ) EM1
13. N\_cells ( $0.1 < dR < 0.7$ ) EM2
14. N\_cells ( $0.1 < dR < 0.7$ ) EM3
15. N\_cells ( $0.1 < dR < 0.7$ ) EM4
16. Sum of cell energies ( $0.1 < dR < 0.7$ ) in EM1
17. Fraction of cell energy ( $0.1 < dR < 0.7$ )/ $dR < 0.7$  in EM4
18. Fraction of cell energy ( $dR < 0.1$ )/ $dR < 0.7$  in EM4

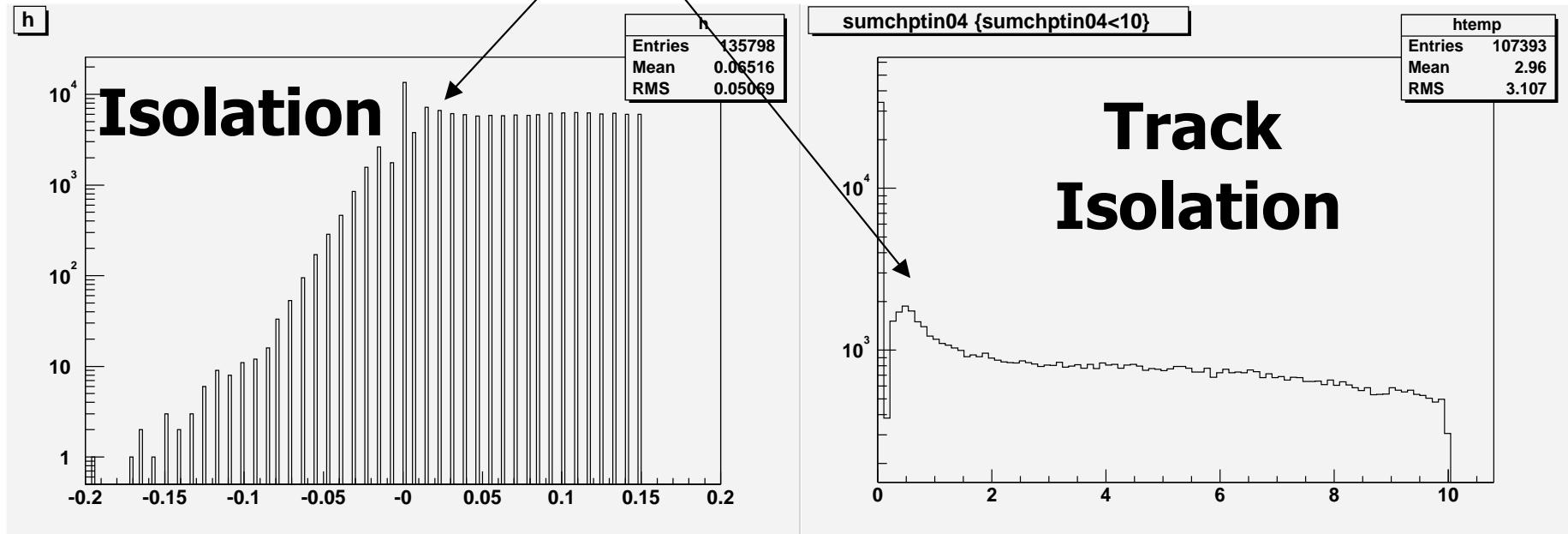
# Looking at 2D correllations and S/B-separation: rphi-width vs Ncells (0.1<|dR|<0.7)

**Signal**



**Background**

# NN may not like discontinuous features of some variables



this one can be fixed in the future  
if more (than 8) bits are  
allocated to store this variable  
in the tmbs

can not be fixed – natural  
consequence of reasonable  
track PT treshold ?

# Summary of NN-optimization road

1. Need to study 2D correlations to find best input variable set
2. Can also use **TerraFerMA** package

# The “Real Life” road:

**EMID Efficiency:  $\sim 86 \pm 2\%$**

- Calculate EMID efficiency wrt  $\text{iso} < 0.15$  using tag/probe object method
- Find NN output cuts that provide same efficiency as current EMID

----- from CCCC -----

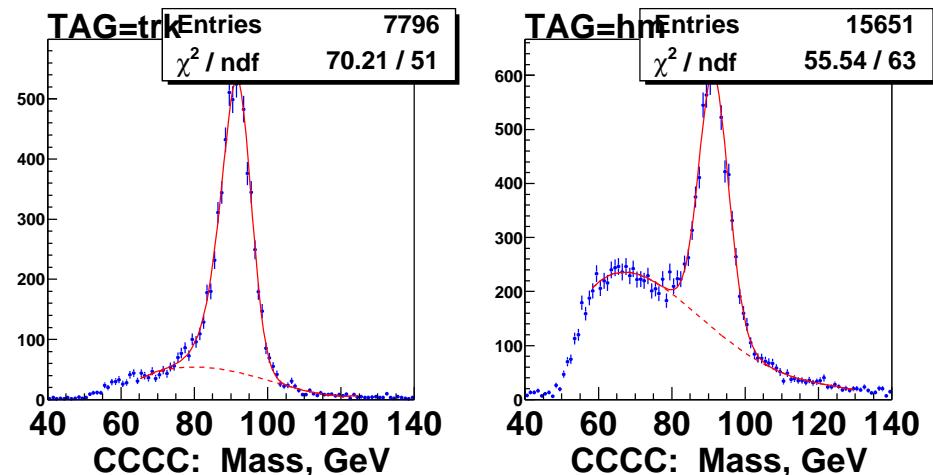
**HMx7 < 10 +  $\text{trkiso} < 2$  GeV**

$88.0 \pm 0.6(\text{stat}) \pm 2.8(\text{syst}) \%$

**NN output > 0.55**

**NN1:  $86.5 \pm 0.6(\text{stat}) \%$**

**NN2:  $89.5 \pm 0.5(\text{stat}) \pm 0.8(\text{syst}) \%$**



----- from CCEC -----

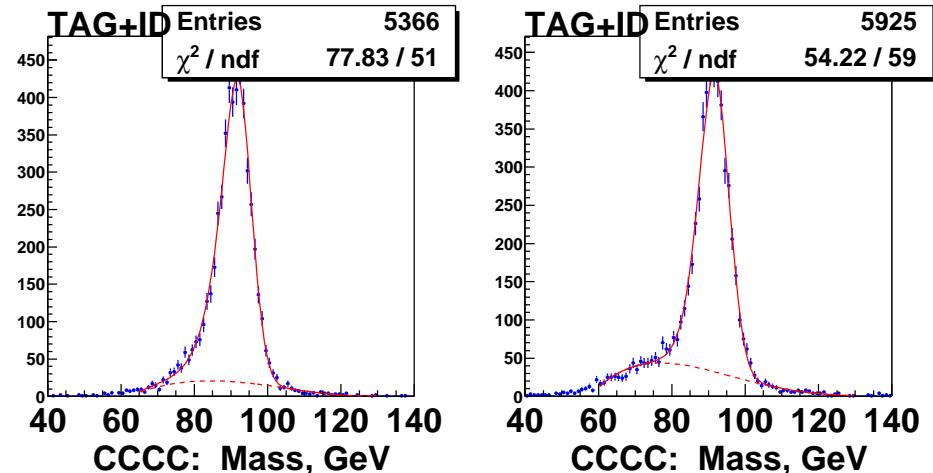
**HMx7 < 10 +  $\text{trkiso} < 2$  GeV**

$85.9 \pm 0.6(\text{stat}) \pm 1.6(\text{syst}) \%$

**NN output > 0.57**

**NN1:  $86.4 \pm 0.6(\text{stat}) \pm 2.1(\text{syst}) \%$**

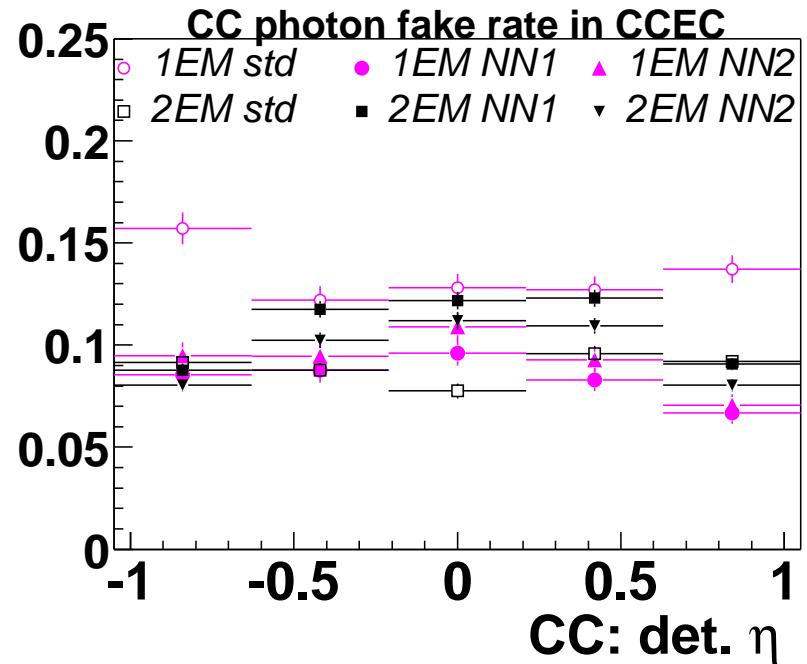
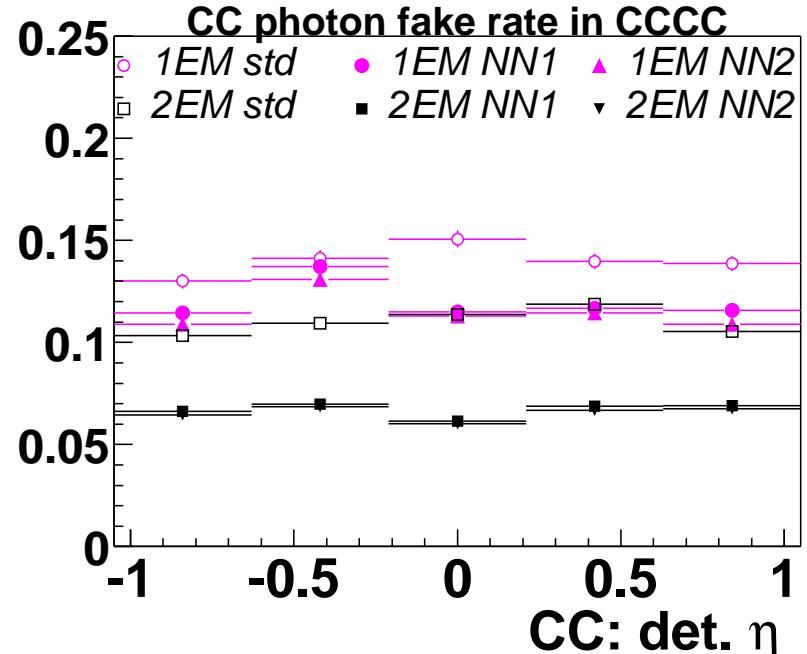
**NN1:  $86.0 \pm 0.6(\text{stat}) \pm 0.8(\text{syst}) \%$**



# Photon Fake Rates

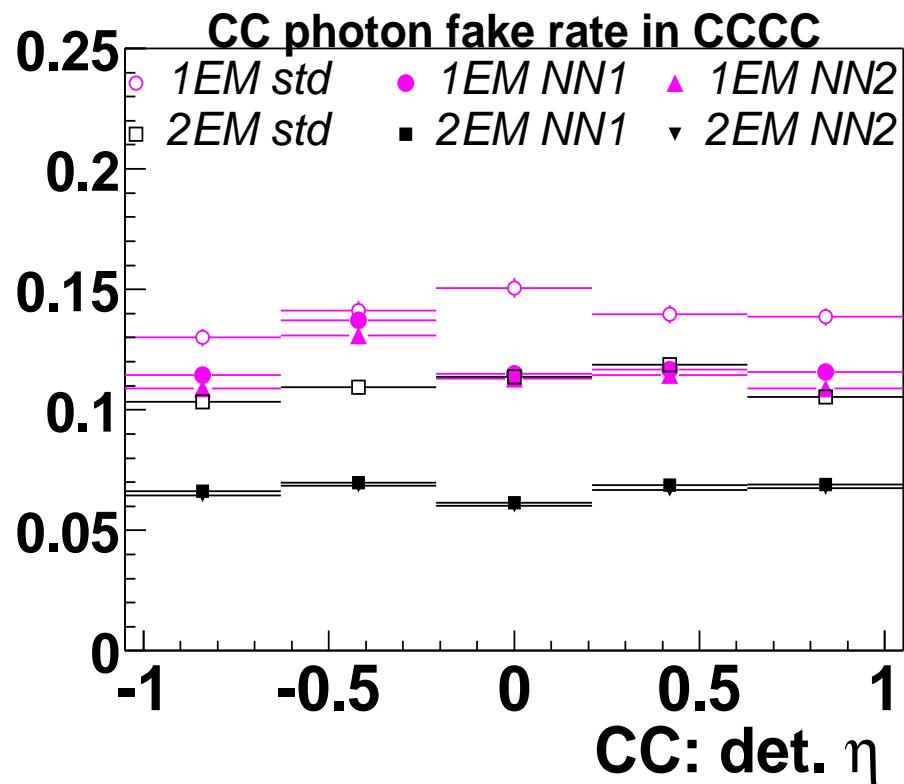
- calculate photon fake rate with established NN output cuts and compare with “HM+trkiso” rates
- look at both Single(exactly one) EM and diEM(Z-mass region vetoed) samples
- compare performance of 2 NNs

**bkg rejection factor of  $\sim 1.5$**

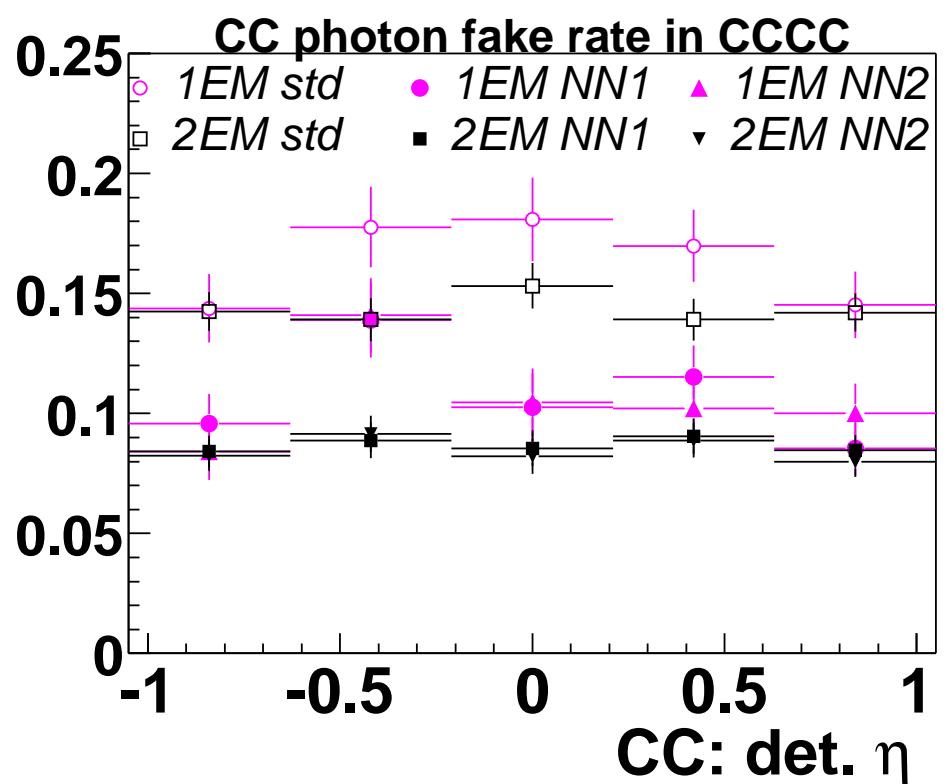


# Photon fake rates (cont'd)

**no diphoton PT cut**



**diphoton PT > 35 GeV**



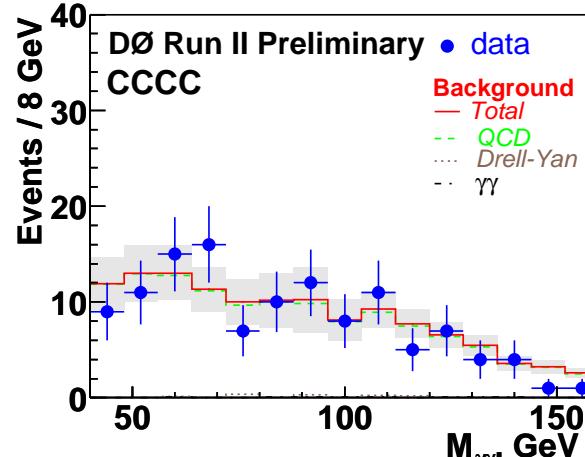
This one is used in the analysis  
to estimate multijet and  
gamma+jet background

# CCCC $\gamma\gamma$ mass

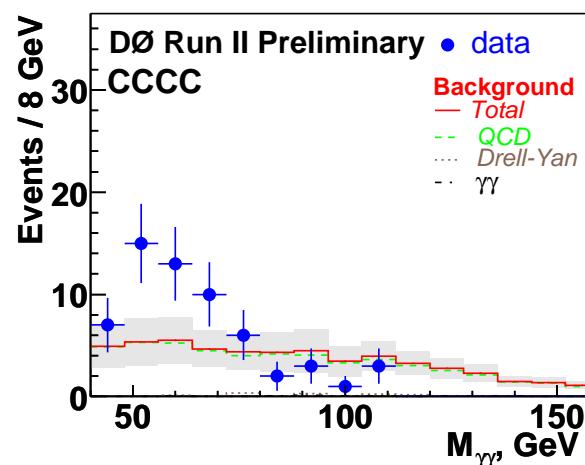
HM7+trkiso  $\rightarrow$

NN1  $\rightarrow$

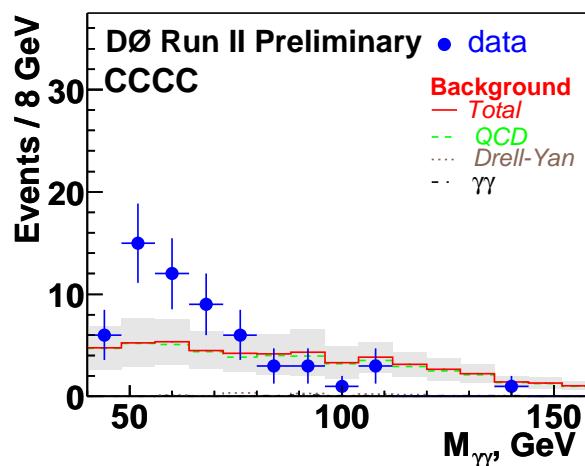
NN2  $\rightarrow$



**data = 121.0**  
**bkgd = 126.3 +- 28.1**  
**QCD = 123.5 +- 28.1**  
**DY = 2.0 +- 1.9**  
 **$\gamma\gamma$  = 0.8 +- 0.2**

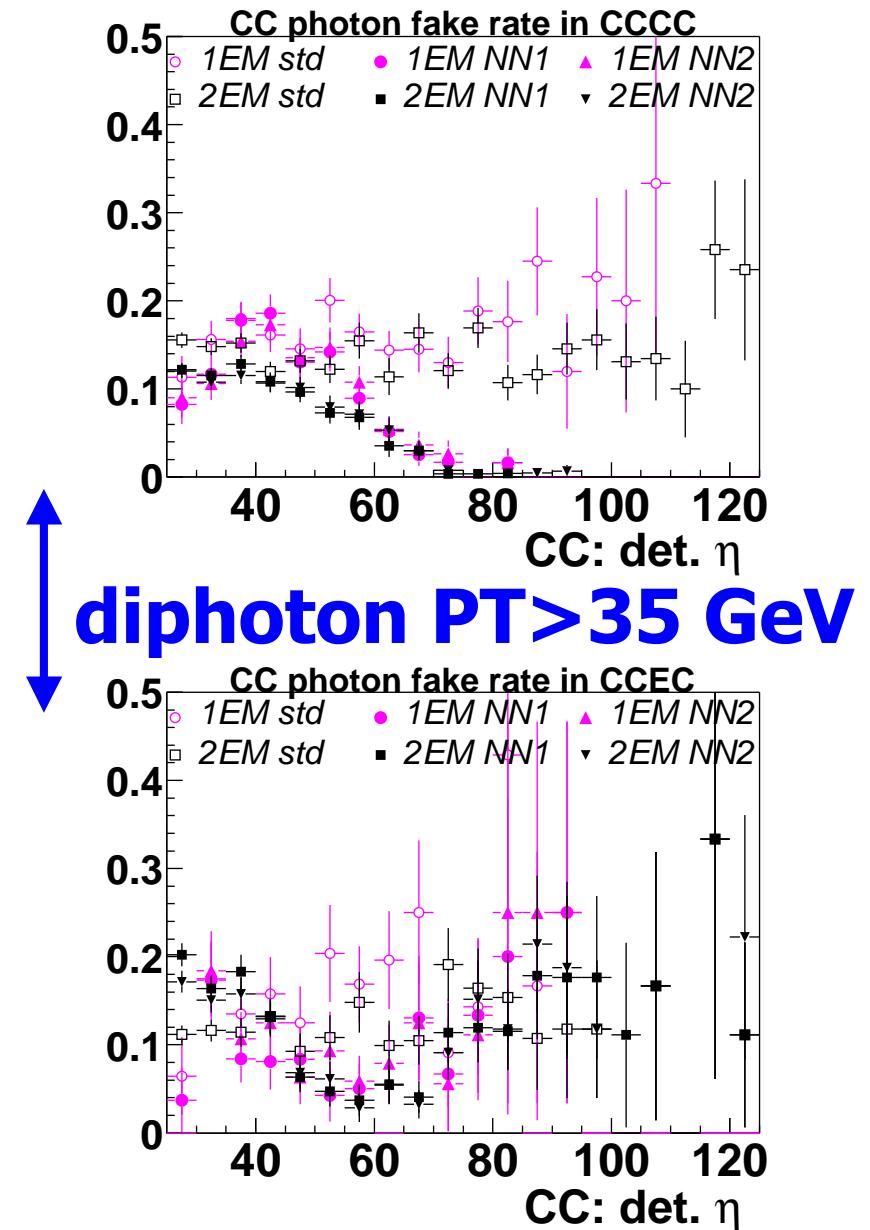
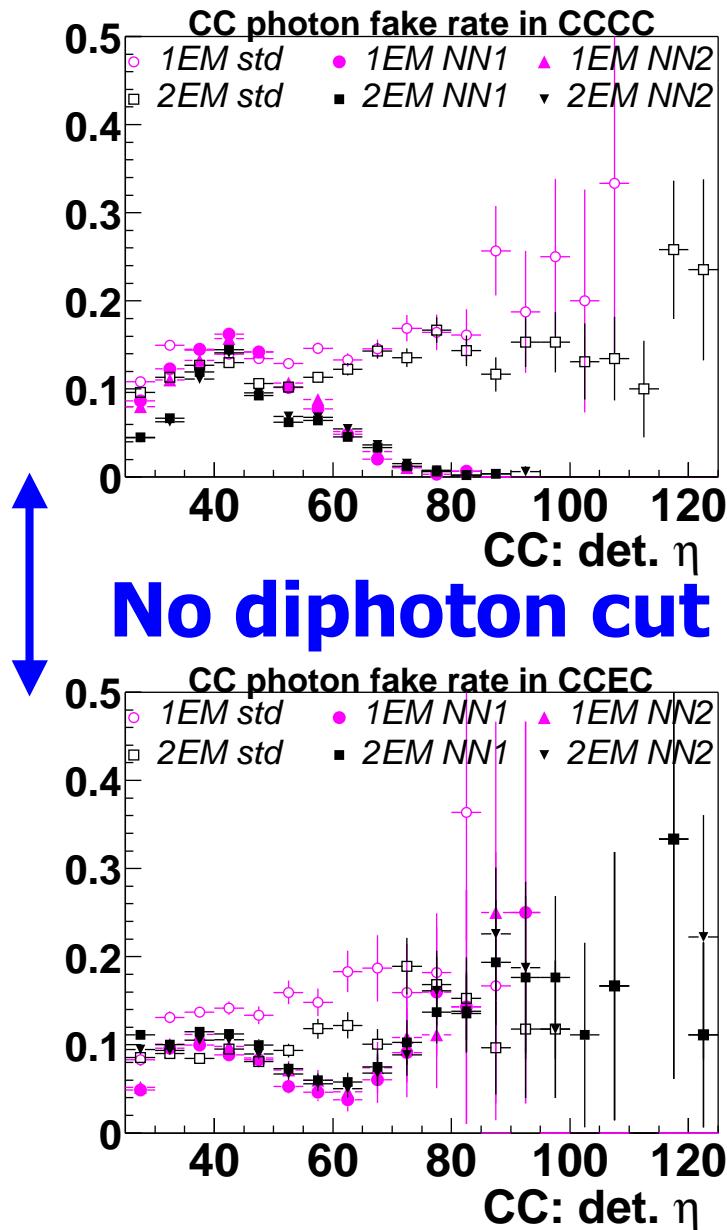


**data = 60.0**  
**bkgd = 53.3 +- 21.2**  
**QCD = 50.6 +- 21.2**  
**DY = 2.0 +- 1.9**  
 **$\gamma\gamma$  = 0.8 +- 0.2**

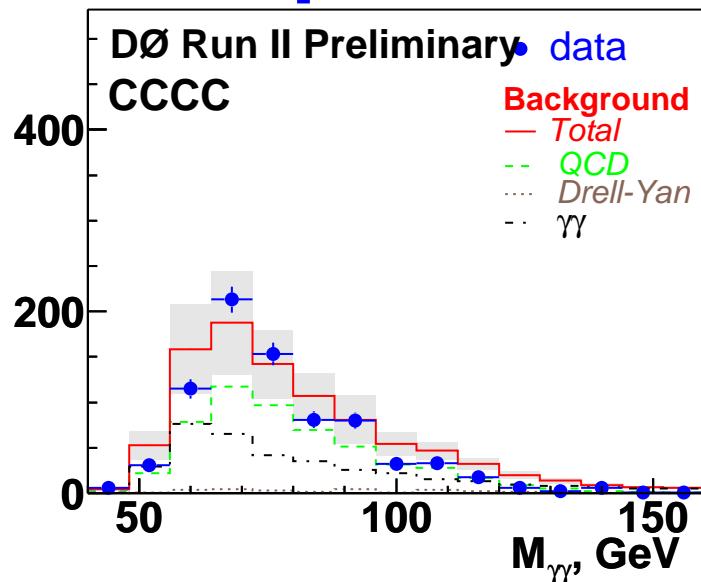


**data = 59.0**  
**bkgd = 51.7 +- 22.0**  
**QCD = 48.9 +- 21.9**  
**DY = 2.0 +- 1.9**  
 **$\gamma\gamma$  = 0.8 +- 0.2**

# Try PT-dependent Fake Rates

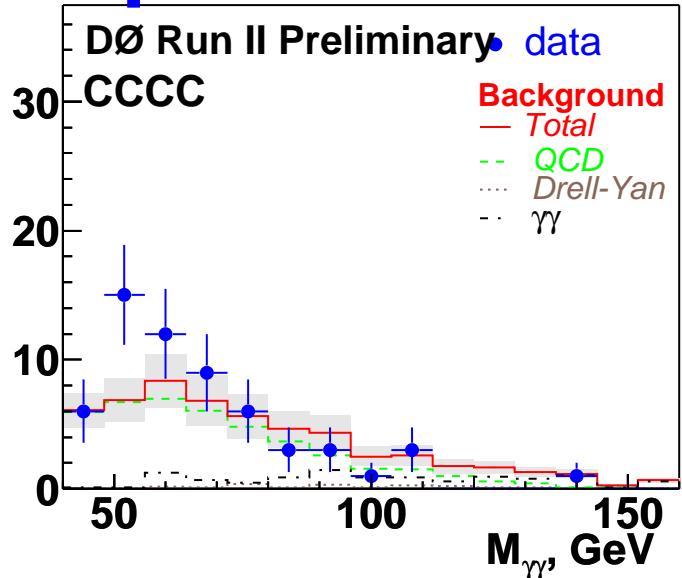


# No diphoton cut



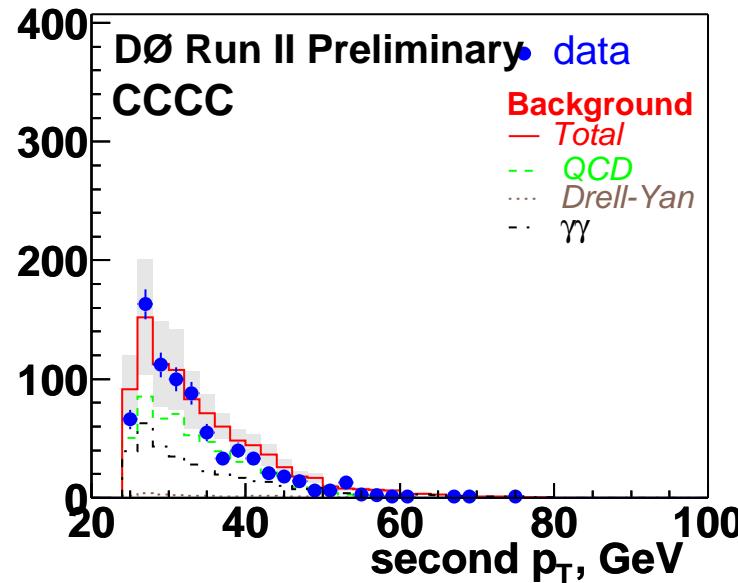
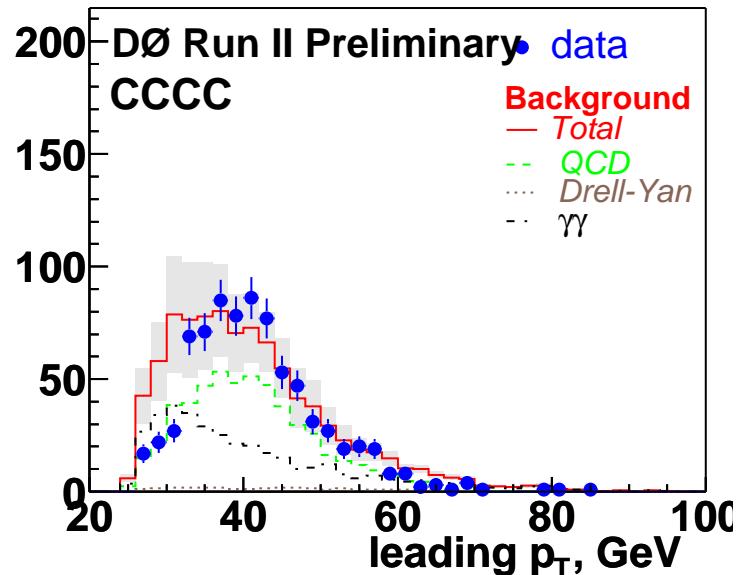
**data = 778.0**  
**bkgd =  $922.9 \pm 242.6$**   
**QCD =  $535.5 \pm 215.1$**   
**DY =  $28.3 \pm 27.9$**   
 **$\gamma\gamma = 359.1 \pm 108.8$**

# diphoton PT>35 GeV

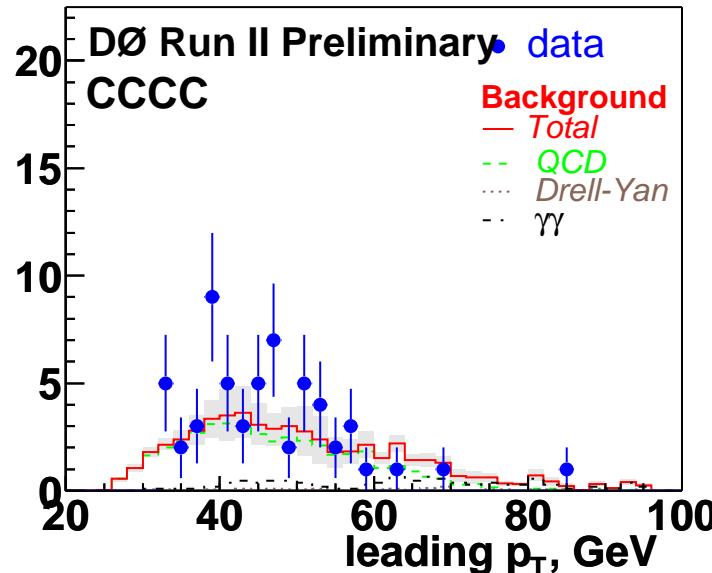


**data = 59.0**  
**bkgd =  $54.5 \pm 14.3$**   
**QCD =  $42.0 \pm 13.8$**   
**DY =  $2.0 \pm 1.9$**   
 **$\gamma\gamma = 10.6 \pm 3.2$**

# No diphoton cut : Photon PT distributions



diphoton  $PT > 35$  GeV: leading Photon PT



# We find that

1. Fake rates with Neural Net  
(unlike with current EMID)  
have strong PT-dependence
2. PT-dependent fake rate allows to describe the  
backgrounds reasonably well
3. Can proceed with the analysis !

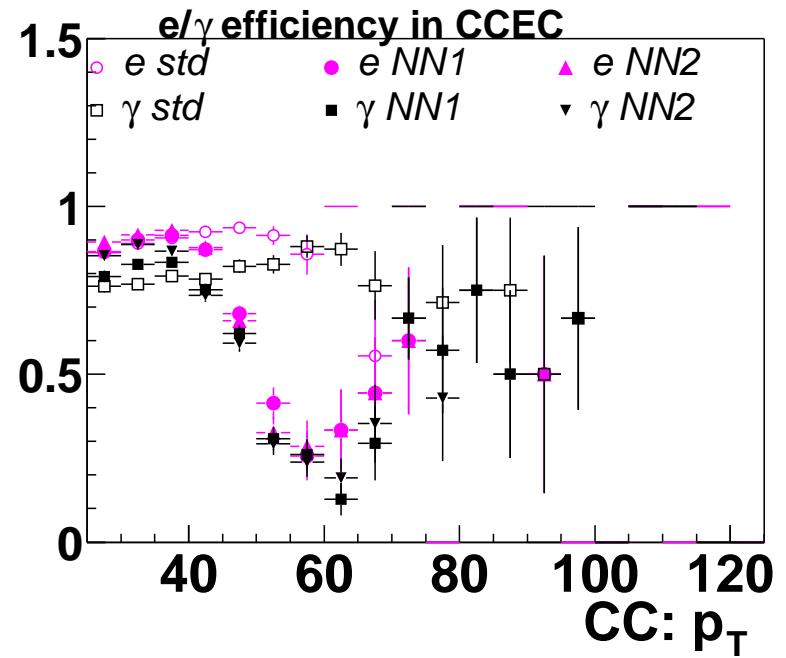
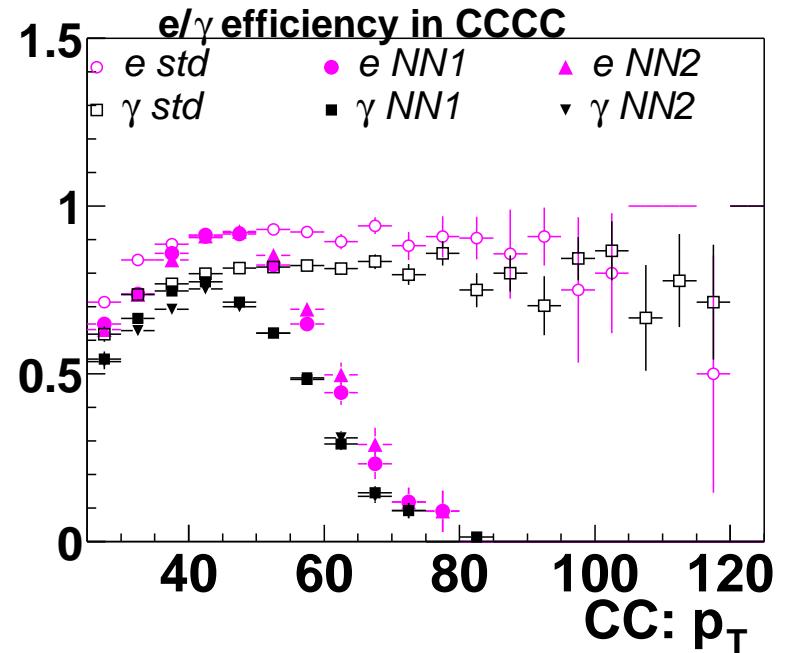
**BUT**

**What about PT-dependence of signal efficiency ?**

**NN signal  
efficiency drops for  
 $\text{PT} >= 40\text{-}50 \text{ GeV}$**

**Won't work for  
 $h \rightarrow \gamma\gamma$  search  
/other high PT  
searches with  
photons**

Is this related to PT-spectrum of training samples ( $Z \rightarrow ee$  data) ?  
*HMx7 (single electron) training sample is flat in PT up to above 100 GeV*



# Try retraining NN with HM7 single electron training sample

- Got the d0sim MC sample (from Michel Jaffre) onto clued0 disk : 70k single electrons
- Encountered 2 technical problems:
  1. d0correct/tmb\_analyze v6/v6a doesn't work with MC anymore (smth seems to have changed in the environment) – tried this with Z->ee and gammagamma MC (Fu tried to reinstall d0correct – didn't help)
  2. D0reco doesn't run on the single electron MC files – smth related to FPS (tried to turn off the FPS via rcps – same complaint)

# **Summary of “using current NN in real life” road**

1. It seems that NN trained on data signal won't work for  $h \rightarrow \gamma\gamma$  / (similar PT range) analysis(e)s  
(due to  $Z \rightarrow ee$  electrons having lower PT than  $h \rightarrow \gamma\gamma$  photons we look for)
2. Try retraining NN with MC electrons covering high PT range and compare background rejection performance. Currently technical problems
3. Other solutions, e.g. finding NN input variable set insensitive to PT-dependence ? Ideas are welcome.